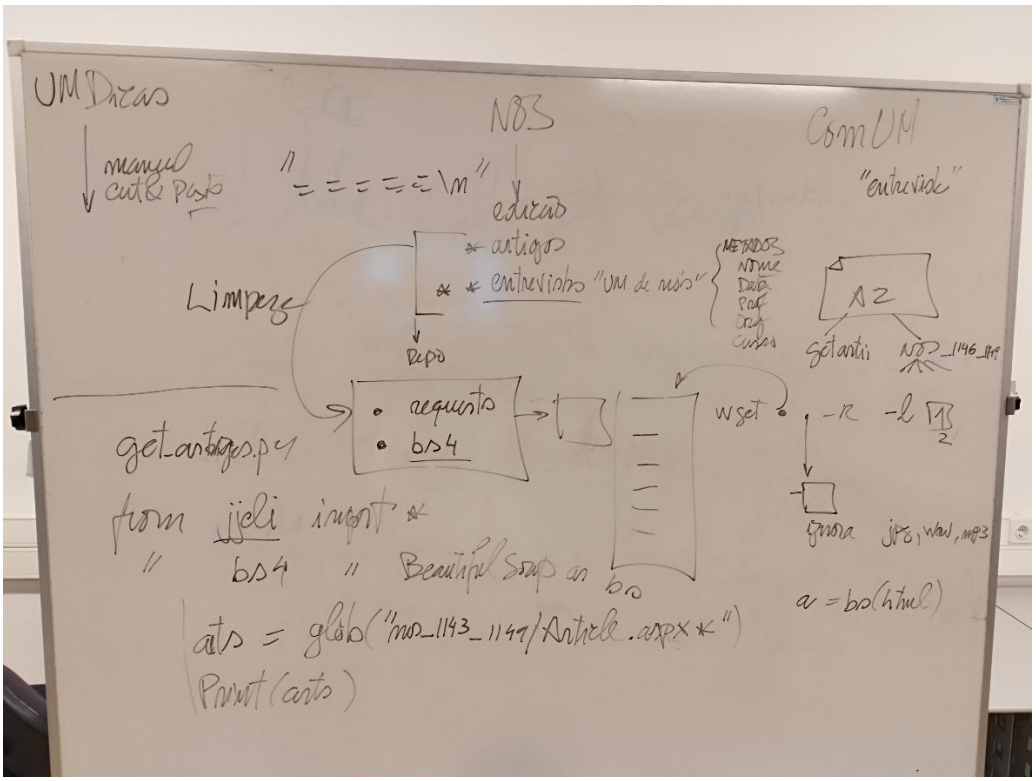
Diário de Bordo

Relativamente ao trabalho de casa da aula anterior, dividimos entre os alunos os artigos do jornal NOS, e cada um ficou responsável pela extração de todos os artigos publicados desde junho de 2015 até dezembro de 2023. Todos estes ficheiros encontram-se no link do Drive do nosso repositório.

O professor colocou no quadro uma explicação do que pretendemos retirar tanto do NOS, como do UM Dicas e do ComUM.



Até ao momento, tivemos mais sucesso na extração de informações do NOS, incluindo edições, artigos e entrevistas "UM de nós". O nosso objetivo é extrair um conjunto específico de dados, como nome do autor, data, profissão, organização, cursos, entre outros. Para isso, é necessário realizar uma limpeza desses dados. Desta forma, o foco principal da aula foi a extração das entrevistas.

Após a devida limpeza dos artigos e entrevistas, podemos colocar no nosso repositório. No caso do ComUM, é necessário identificar um separador para as “entrevistas”. Quanto ao UM Dicas, podemos usar um webscrapping mais “manual”, cut&paste. Durante a aula, observamos que este tem uma seção com 21 páginas de entrevistas.

Testes realizados:

Uma vez que utilizamos o *winget* para obter os dados dos links, não precisamos do *Requests*, mas o *bs4* (Beautiful Soup) é essencial para limpar os dados extraídos (artigos e entrevistas).

Começamos, então, por extrair os artigos de uma das pastas ZIPs e criamos um programa Python a que demos o nome de "get.artigos.py", usando o bs4 para separar as informações necessárias.

Nesta aula, utilizamos a pasta “nos_1146_1149” e começamos por escrever o seguinte código:

```
from jjcli import *
from bs4 import BeautifulSoup as bs


ats= glob("nos_1146_1149/Article.aspx*")
print (ats)
```

Fizemos então o *import* do *jjcli* e do *bs4* e, de seguida, utilizamos a função ‘glob’ para buscar todos os arquivos no diretório “nos_1146_1149” que começam com "Article.aspx" e imprimimos o resultado. O resultado é uma lista contendo os nomes de todos os arquivos nesta pasta que têm este padrão.

De seguida, selecionamos um ficheiro específico para analisar e entender que informações poderíamos extrair. Escolhemos o arquivo "Article.aspx@id=3669", que corresponde a uma das entrevistas da seção "UM de nós". Abrimos o ficheiro no VS Code para examinar o HTML e identificar os dados relevantes.

Ao utilizar , tentamos extrair o conteúdo completo do artigo com o seguinte código:

```
def proc_article(html):
    #print (len(html)) #esta a contar os carateres de cada artigo
    a=bs(html) # cria uma árvore documental
    art= a.find("div", id="artigo") #procura no html
    print ("=========\n", art.get_text()) #get_text - Retira apenas o texto mesmo, sem html

for file in ats:
    with open(file, encoding="utf-8") as f:
        html= f.read()
    proc_article (html)
```

Do código presente, dá-se especial destaque ao “a.find” ou “a.find_all”, para procurar no HTML o que precisamos e "get_text" para apresentar o conteúdo desejado sem o HTML, apenas o texto.

De uma forma geral, nesta aula aprendemos como fazer limpeza dos dados e de como extrair a informação que pretendemos.

Realizado por: Andreia Gonçalves, pg51914